

### AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions of claims in the application:

#### **Listing of Claims:**

1. (Currently Amended) A system stored in computer memory that facilitates building an application using a development framework, the system comprising the following computer executable components:

an exposor component that exposes a set of classes, which set includes at least one of a framework class of the framework and a project class of a project, ~~and~~ which at least one of the framework class and the project class is used to develop the application[[.]] ; and

the exposor component further comprising an identifier component that identifies from source code one or more members of at least a group class and a compiler that compiles the one or more members into the at least one group class thus creating a namespace that provides access to one or more classes that are used more frequently than other classes.

2. (Original) The system of claim 1, the set of classes includes at least one of a class related to a computing device on which the application will be run, a class that provides information about the application, an object that provides information about a user that runs the application, and a class that is commonly used in the project.

3. (Original) The system of claim 2, the class that is commonly used is related to one of a form, a web service, a resource, and a setting.

4. (Original) The system of claim 1 facilitates creation of a single entry point to common classes for building the application.

5. (Original) The system of claim 1, the exposor component exposes a class of a plurality of namespaces of the framework.

6. (Original) The system of claim 1, the exposers component facilitates creation of a namespace that provides hierarchical access to instances of classes that are commonly used to develop the application.

7. (Original) The system of claim 6, the namespace includes a default set of the classes.

8. (Canceled)

9. (Original) The system of claim 1 is extensible such that a new class can be exposed that is provided in accordance with at least one of an expansion of the framework and an improvement to the framework.

10. (Canceled)

11. (Original) A computer employing the system of claim 1.

12. (Original) The system of claim 1, the set of classes is a top-level set that includes one or more classes related to the application, a computer running the application, a user running the application, a form of the project, a web service referenced in the project, a resource of the project, and a setting of the application.

13. (Currently Amended) A system stored in computer memory that facilitates building an application within a development framework, comprising the following computer executable components:

a compiler that compiles code; and

an identification component that receives search information related to class information of at least a class to be identified, which identification component signals the compiler to search the code based on the search information and tag the class information to dynamically generate the class comprising one or more members identified from source code hence providing hierarchical access to instances of classes that are used more frequently than other classes.

14. (Original) The system of claim 13, the compiler tags the class information during compilation of the code.

15. (Original) The system of claim 13, the compiler provides user access to the tagged information.

16. (Original) The system of claim 13, the class information is tagged utilizing a compiler attribute.

17. (Original) The system of claim 13, the tagged class information is pulled out and compiled separately with respect to compiling the code.

18. (Currently Amended) The system of claim 13, the class ~~is generated dynamically,~~ ~~and~~ includes strong types and bounded access that points only to an object of the class.

19. (Original) The system of claim 13, the system dynamically generates types in a namespace that reference internal resources.

20. (Currently Amended) A system that facilitates building an application within a development framework stored in computer memory, the system comprising the following computer executable components:

a compiler that compiles code, which compiler receives search information associated with class information from an identification component, searches the code based on the search information, ~~and~~ tags the class information and dynamically generates a class that refers to an internal resource by pulling out the tagged class information and compiling the tagged class information.

21. (Canceled)

22. (Original) The system of claim 20, the compiler compiles the tagged class information to generate a class that facilitates user access to an internal resource.

23. (Currently Amended) A method of aggregating functionality in support of building an application, comprising:

identifying a class of objects to be returned from source code;

searching the source code for one or more of the objects;

collecting the one or more objects that are found;

generating a property for each of the one or more objects that are found; ~~and~~

accessing the one or more objects that have the associated property[[]] ; and

compiling the one or more objects that are associated with a given property, into the class.

24. (Canceled)

25. (Original) The method of claim 23, further comprising associating the class with an indicator that is unique to the class.

26. (Original) The method of claim 23, the one or more objects that are found, are collected according to an attribute.

27. (Canceled)

28. (Currently Amended) A computer-readable medium having computer-executable instructions for performing a method of building an application, the method comprising:

providing a programming language compatible with an application environment, the language used for developing the application;

identifying objects of the application environment;

searching source code of the application environment for one or more of the objects;

generating a property for each of the one or more objects that are found;

accessing the one or more objects that have the associated property[[]]; and

compiling the one or more objects into a class associated with a namespace.

29. (Canceled)

30. (Canceled)

31. (Original) A system that facilitates building of an application, comprising:  
means for identifying an object of an application development environment;  
means for searching source code of the environment for one or more of the objects;

means for generating a property for each of the one or more objects that are found;

means for returning the one or more objects that have the associated property;

means for compiling the one or more objects into a class; and

means for associating the class with a namespace.

32. (Original) The system of claim 31, the means for compiling fetches source files from a runtime library.

33. (Original) The system of claim 31, further comprising means for injecting source code into a user project based on a library that was referenced.

34. (Original) The system of claim 31, the one or more objects are top level objects that have a class declaration associated therewith.

35. (Original) The system of claim 31, the property is part of source code that is embedded in a runtime dynamic linked library as a resource.

36. (Original) The system of claim 35, the means for compiling automatically references the library, and checks for the presence of the resource for all compilations.

37. (Original) The system of claim 35, the means for compiling adds contents of the resource as a hidden source file buffer to a project defined within the environment.

38. (Original) The system of claim 31, the means for compiling uses attribute arguments to collect class members of a group of the one or more objects to generate underlying code of the group.

39. (Original) The system of claim 31, further comprising means for employing a number of top-level classes according to the application being developed.